ACE: A digital Floating Point CNN Emulator Engine

Béla Fehér+, *Péter Szolgay, Tamás Roska, András G. Radványi, Tamás Szirányi, Márton Csapodi, Károly László, László Nemes, István Szatmári, Géza Tóth, Péter L. Venetianer*

Analogical and Neural Computing Systems Laboratory
Computer and Automation Institute, Hungarian Academy of Sciences
P.O.B.63, H-1502 Budapest,Hungary
+ Measurement and Instrument Engineering Department, TU Budapest
H-1521 Budapest, Müegyetem rkp.9, Hungary

ABSTRACT: *The architecture of ACE, a multiprocessor analogic CNN emulator engine consisting of 2 to 16 TMS320C40 floating point DSPs is introduced. The engine containing up to 512 Mbyte RAM (enough to store a 512*512*512 sized CNN cube) can be controlled through its SCSI port. It can either accelerate the multilayer CNN simulator CNNM or be accessed directly from the high level, C-based analogic CNN language ACL to achieve the simulation speed of ~2.8μsec/cell/iteration/DSP for 3*3 linear templates.*

1. Introduction

Cellular Neural Networks [1] are cellular, analog, programmable, multidimensional processing arrays with distributed logic and memory. The processing elements are locally connected. The CNN architecture can be programmed with the interconnecting weights - templates - of processing cells. The extension of the CNN paradigm is the CNN Universal Machine [2] the first stored-program, analogic spatio-temporal array computer, where distributed and global memories together with logic is used to implement complex analog+logic=analogic CNN algorithms. The analog VLSI implementations of the CNN provide tera operations per second computing speed with 6-8 bit accuracy.

Digital CNN simulators [5] can be used in the design and testing analogic algorithms.

Large dynamical systems can be analysed by the CNN where the transient behaviour have to be computed [7]. In such applications the accuracy of the results are critical, meaning that the integer representation of the numerical values (of input, output, state and templates) in the simulation is not enough.

Although the analog VLSI implementations of CNN could exhibit a supercomputer speed in the analysis of dynamical systems, the CNN arrays that the presently available analog VLSI technology can produce contain "only" some 1000 cells, and the accuracy of the analog implementation is limited. A digital VLSI hardware emulation seems to be a good compromise when higher accuracy and propagating templates on a larger array are essential. In good many applications the fixed point digital hardware CNN accelerator/emulator board (HAB) with 1 million virtual CNN processor [3, 5] proved to be a powerful tool, but in solving some types of partial differential equations the accuracy was not enough. Therefore it was decided to build up the Analogic CNN Emulator Engine (ACE), a new emulator based on floating point digital signal processors. In this paper we report on this Engine.

2. The architecture

The ACE accelerator engine is connected via SCSI to a PC or a workstation, where the data and results of the simulation are stored and displayed. Large dynamical systems, such as a CNN cube consisting of 512*512*512 cells in case of maximum ACE configuration, can be computed with four-byte floating point accuracy in quasi real time. In addition to the improved accuracy, the computing speed is ~2.8μs/cell/iteration/processor, more than three times higher then that of the fixed point emulator system HAB.

The architecture of the ACE is based on the optimal utilisation of the available processing power of the built-in TMS320C40 floating point DSP units. The maximum 50 MFLOPS/processor arithmetic performance could be significantly smaller, if the necessary operands were not readily available during the calculation.

The DSPs are to calculate the discretized state and output equations (1) of the CNN. Namely, in the n-th time step the new, $(n+1)$-th state value of the ij-th element v_{xij} can be calculated from the stepsize h and the old n-th state v_{xij} and the output v_{yij} values of the neighbouring cells as function of the state; A and B are the cloning templates, $N_r(ij)$ is the r-neighbourhood of the ij element. Function f is a piecewise linear sigmoid of unity slope.

0-7803-3261-X/96/$5.00 © 1996 IEEE.

$$v_{xij}(n+1)=(1-h)v_{xij}(n)+h\,[\sum_{C_{kl}\in N_r(ij)} A(ij;kl)v_{ykl}(n)\;+\sum_{C_{kl}\in N_r(ij)} B(ij;kl)v_{ukl}(n)+I\,] \qquad (1)$$
$$v_{yij}(n)=f(v_{xij}(n))$$

For efficient computation, the selected architecture implements two hierarchical levels of parallelism. On the upper (cell) level, the multiprocessor, data type parallelism is applied [3], as the new values of the individual elements can be calculated independently from each other. The computational power on this level is directly proportional with the number of the physical processors. Since this number is always significantly smaller than the number of cells (virtual processors) to be computed, careful mapping should be devised to reduce the interprocessor communication. Horizontal band decomposition of the 2D input data structure can minimise the interprocessor communication through the upper and lower borders. On the lower, or operation level, the functional type parallelism is used, directly supported by the complex pipeline architecture of the TMS320C40 DSPs. The internal parallel floating point multiplier and arithmetic unit, the address generators, the large set of separate data buses completed with external elements ensures high CPU performance. The two parallel levels of the architecture are connected only through the necessary interprocessor communication between the border cells and their neighbourhoods. This communication is realised through the high speed communication channels of the DSPs, and is handled by the independent DMA coprocessors.

The ACE system works as a slave emulator engine. To exploit its computational power in a general computer environment, a high speed, flexible and versatile communication interface was necessary. It was to provide a simple connection to any type of host machines and easy mechanism for bulk data transfer. Evaluating the requirements, the widely accepted SCSI interface has been implemented, with provision to the 16 bit wide synchronous data transfer mode.

The general architecture of the ACE is shown in Fig. 1. The modular system consists of the SCSI interface unit and the computational units. Each computational unit contains two TMS320C40 DSPs with the appropriate memory modules and control logic. The maximum number of the 'C40 DSPs in the system is 16, so the theoretical computational performance limit is 800 MFLOPS. The main data memory modules are configurable up to 32 Mbyte/processor for large pictures or input data arrays.

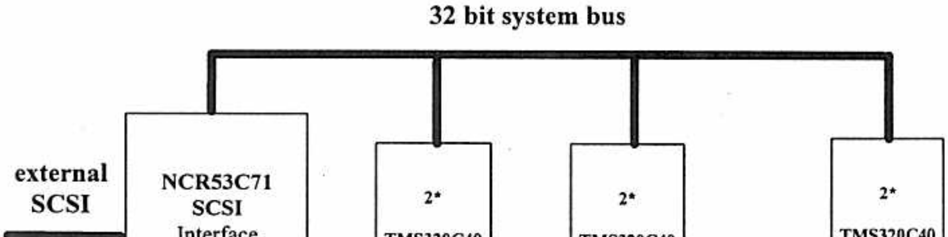


Figure 1: The architecture of the CNN-ACE simulator engine

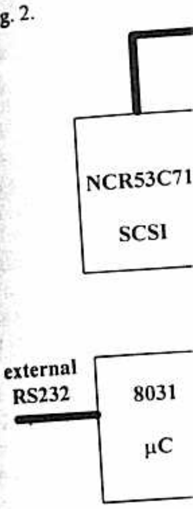
The physical design considerations are as follows. The emulator engine is installed into a standard midi size IBM-PC rack. This off the shelf component selection reduced the cost, limited the design efforts to the PCBs, and provided proven mechanical construction. The SCSI interface unit has been realised as a motherboard with 8 extension slots for the DSP boards of computational units, each of them with two 'C40s. The DSP boards are standard, full size IBM PC add-in boards. Four of the six communication channel of the 'C40 processors are used, from which one connects the processors on the same board, while the other three are for external connections. Using these extension channels different kinds of multiprocessor configuration patterns can be realised on the upper parallelization level. External devices, for example video camera input or display output can be connected, as well.

3. The hardware block

The ACE hardware archi
the most important features v

3.1. The SCSI interface ar

The emulator uses its pri
type, target unit on the SCS
communication processor.
capable to handle all the
operation, without the sup
program located in its ow
according to the applicatio
buffers or modified, config
in Fig. 2.



Both buses of the S
synchronous and the n
of the host machine. I
with the DSP units. Th
slot defines a unique b
as addresses for contr
system there is a glob
operation of the multi

Basically the syste
the SCSI processor is
is not active) the SC
executes the power-
processor can mana
errors or serious SC
supervisor determin
sets the control to t
supervisor communi
provides high priori

The supervisor
with the necessary
during normal ope
provides a test acc

3. The hardware blocks

The ACE hardware architecture follows the main block diagram in Fig. 1. In this hardware description only the most important features will be discussed.

3.1. The SCSI interface and supervisor controller unit

The emulator uses its primary SCSI interface for control and data transfer. The emulator is simple processor-type, target unit on the SCSI bus. The interface has been implemented using the NCR53C71 intelligent SCSI communication processor. The main advantage of this program-controlled SCSI processor unit is, that it is capable to handle all the necessary SCSI bus management and data block transfer phases during normal operation, without the supervisor processor intervening. Its operation is controlled by its unique SCRIPT program located in its own program memory. The instruction set of this SCRIPT language is optimised according to the application environment. The SCSI processor has a RAM memory also for temporary data buffers or modified, configuration specific SCRIPT programs. The block diagram of the SCSI interface is shown in Fig. 2.

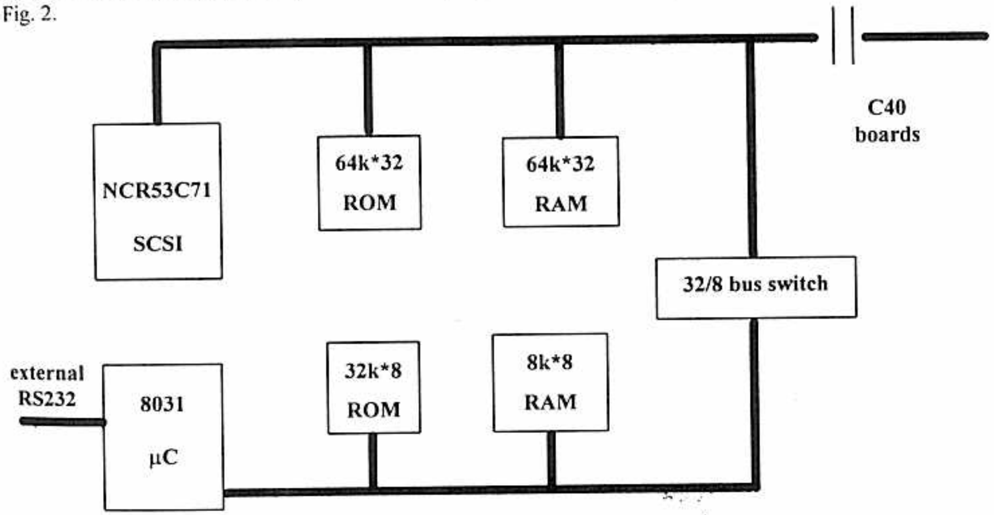


Figure 2 : The SCSI processor and the supervisor controller

Both buses of the SCSI processor are configurable. On the SCSI bus side, the device offers the high speed synchronous and the new 16 bit wide data block transfer methods too, their application depends on the capability of the host machine. Internally the processor implements a 32 bit wide synchronous data bus to communicate with the DSP units. The resources of the DSP units are directly available from the system bus. Every expansion slot defines a unique base address for the DSP boards, determining identification numbers for the DSPs as well as addresses for control and status registers and the memory modules. Above the unified control of the whole system there is a global control register. This duplicate control mechanism allows not only the synchronised operation of the multiprocessor DSP system, but makes possible the individual processor control as well.

Basically the system has two operation modes: the supervisor and the user mode. In the normal or user mode the SCSI processor is the master of the CNN-ACE emulator system. After power-up (if the supervisor interface is not active) the SCSI processor checks the system configuration, the number of the implemented DSP units, executes the power-on system test and waits for the SCSI selection and commands from the host. The SCSI processor can manage the system operation perfectly, unless there is no fatal error situation. In case of fatal errors or serious SCSI exceptions, it halts its operation and waits for the help of the supervisor controller. The supervisor determines the source of exception by directly reading the NCR53C71 internal status registers and sets the control to the appropriate exception handling SCRIPT routine. If at the power-up or after reset the supervisor communication interface is found alive, the supervisor controller takes the control of the system and provides high priority right to the host.

The supervisor controller has been implemented with the Intel 8031 8 bit microcontroller. It is completed with the necessary 32bit/8bit bus transfer switch. Its limited performance has no effect on the emulator system during normal operation, because in that case it is not active. In supervisor mode the 8031 microcontroller provides a test access port on its asynchronous serial I/O. The main advantage of this additional port has been exploited during the development phase of the CNN-ACE. Simple monitor program (completed with a high priority user interface on the host) was used to check the functionality of units, to load and start test programs, investigate low level system variables, etc.

3.2. The TMS320C40 DSP boards

The architecture of the extension boards is designed according to the computational task given in (1). The main computational workload is the calculation of the double (or in case of constant v_u input values the single) inner product of (1), and the non-linear mapping of function *f*. The inherently large computational performance of the C40's is achievable only, if there is no internal pipeline or resource conflicts during execution. The C40's guarantees only the correct execution, but not an optimal one. The most frequent pipeline conflicts are the address register and memory conflicts [6]. For efficient operation during the internal loop of the calculation of (1) care must have been taken to avoid these conflicts and provide the best utilisation of available resources.

In the CNN-ACE multiprocessor emulator engine a distributed memory architecture has been implemented. Each processor has its own memory modules. Interprocessor communication is performed only via 'C40s communication ports. The architecture of one DSP unit is shown in Fig. 3. Each board contains two equivalent DSP units.

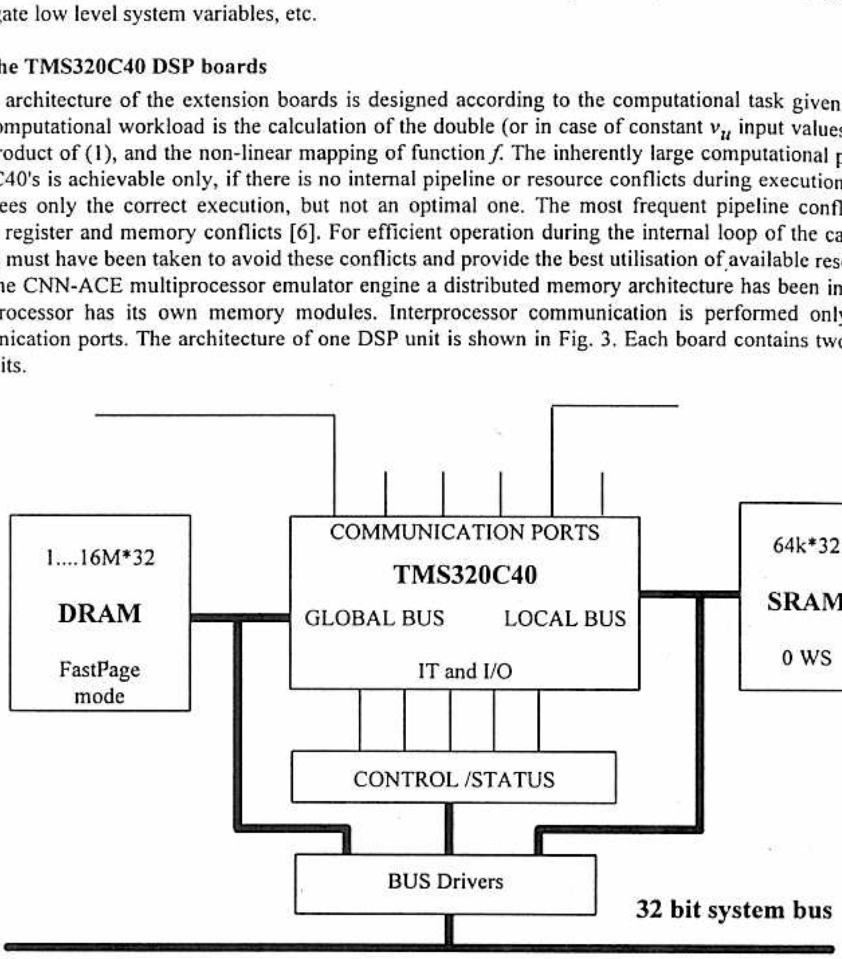


Figure 3: The architecture of the TMS320C40 DSP unit

The DSP modules have two operating modes. In the bus access mode, the 'C40 processor is in the reset or idle state and the memory modules are accessible from the 32 bit system bus. In the processor mode, the unit is detached from the system bus and the processor has access to the memories. Independent memory controllers are implemented to achieve the best possible timing parameters in both operating modes. The control/status logic ensures the proper switching between the two modes and controls the operation of the processor. The control command set includes the usual RESET, RUN, STOP, CONTINUE commands. Accordingly, the processors can be in the RESET, RUN, STOP and WAIT states. The status registers contain information on the status of the processors, and the global system, as well. The global status information is collected by wired-or global status lines on the system bus.

Each processor has two types of memory banks, according to the application requirements. On the C40's global bus there is a large capacity DRAM module, its size can vary from 4 Mbyte to 32 Mbyte. This memory is used to store the input and output data arrays. During computation it is accessed only by the DMA coprocessors to move the current data blocks into the high speed internal memory or to the local bus SRAM. To reduce the number of wait states during DMA transfers the processor mode DRAM controller supports the Fast Page access mode.

The local bus SRAM module serves as program memory and temporary storage for calculation results. According to the analysis of the 'C40 internal bus structure, maximum 4 memory accesses can be processed at the same time, from which one external access is a program fetch, the other is a DMA cycle. The internal

pipeline operational units can co
or destination) needs external m
Each TMS320C40 processo
global 40MHz system clock. In
multiprocessor system is ensure

4. The software compon

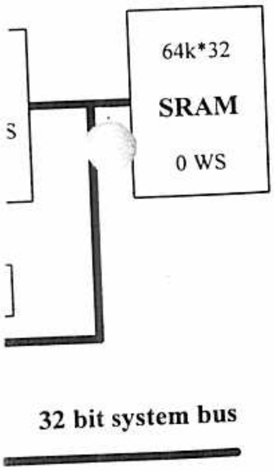
Including its host, several
one type - the TMS320C40 - c
whilst communicating with th
The main role of the ho
environment receiving user c
and storing results. Since CN
exists a menu controlled deve
providing functions as memo
options. Both host programs
serial port, a full-fledged
troubleshooting. The actual
inside the communication la
Analogic CNN Language A
The CNN-ACE motherb
microcontroller. Three proc
and receive data block. Th
language format with the S
either be downloaded into a
microcontroller EPROM to
was enough to program on
The role of 8031 micro
cases it is its task to react
8031 Macro Assembly L
providing complete acces
monitor services are direc
Daughter-boards conta
ACE, each of them with
power-up test, programs
communication channel.
corresponding RESET-R
Assembly Language is
Assembly Language c
START_ADDRESS loa
available for application
The CNN Simulato
controlled from the ho
manually optimised and
i) template routines to c
image segment frames,
local control and statu
optimised routines are

4.1. Application exa

For illustrating the
40*40*3 CNN model
snapshot of the twist
consisting of 40*40*(
in achieving correct r
was 7.7μsec/cell/itera

ɔrogram (completed with a high
;, to load and start test programs,

ıputational task given in (1). The
ɔnstant v_u input values the single)
large computational performance
flicts during execution. The C40's
'requent pipeline conflicts are the
internal loop of the calculation of
lisation of available resources.
ırchitecture has been implemented.
tion is performed only via 'C40s
Each board contains two equivalent



'40 DSP unit

ɪe, the 'C40 processor is in the reset or
bus. In the processor mode, the unit is
es. Independent memory controllers are
erating modes. The control/status logic
operation of the processor. The control
mands. Accordingly, the processors can
contain information on the status of the
on is collected by wired-or global status

application requirements. On the C40's
m 4 Mbyte to 32 Mbyte. This memory is
accessed only by the DMA coprocessors
ır to the local bus SRAM. To reduce the
l controller supports the Fast Page access

emporary storage for calculation results
ı 4 memory accesses can be processed at
the other is a DMA cycle. The internal

pipeline operational units can complete the execution of the instructions in a single cycle, if one operand (source or destination) needs external memory access only [6], calling for a careful selection of operand ordering.

Each TMS320C40 processor can operate from its own, independent 40 MHz clock generator or from the global 40MHz system clock. In the latter case, after a proper RESET sequence, the full synchronisation of the multiprocessor system is ensured.

4. The software components of CNN-ACE

Including its host, several processors of four different types cooperate when CNN-ACE is in action; and of one type - the TMS320C40 - can be as many as 16. Each processor has its own program to perform distinct tasks whilst communicating with the others in due course.

The main role of the host program is to provide a user interface, a menu controlled CNN simulation environment receiving user commands, requests and data, monitoring the progress of processes, and showing and storing results. Since CNN-ACE can also be used as a general purpose multiprocessor accelerator, there exists a menu controlled development environment too, opening also low level access to the ACE resources; by providing functions as memory read/write, program download and run, together with several monitoring and test options. Both host programs incorporate a fast communication layer towards CNN-ACE SCSI port. Through the serial port, a full-fledged slow communication channel is also available for testing, debugging and troubleshooting. The actual channel is selected with a simple switch, the operational differences are all hidden inside the communication layer. The host programs are written in Microsoft C/C++, and extensively use the Analogic CNN Language ACL[8].

The CNN-ACE motherboard accommodates an NCR53C720 SCSI bus controller and its supervisor 8031 microcontroller. Three procedures are implemented to drive the SCSI processor: initialisation, send data block and receive data block. These procedures are written in NCR SCRIPT Language, and converted into a C language format with the SCRIPT Compiler. Embedded in a C language control program, the SCSI routines can either be downloaded into a SCSI RAM or stored in its ROM. In CNN-ACE these routines are stored in the 8031 microcontroller EPROM together with its programs and loaded into the SCSI RAM after power-up. This way it was enough to program one ROM only instead of two.

The role of 8031 microcontroller in normal operation is limited to power-up procedures, while in exceptional cases it is its task to reactivate the SCSI communication channel. The microcontroller programs are written in 8031 Macro Assembly Language and consist of hardware tests, SCSI exception handling and a monitor providing complete access to all parts and resources of CNN-ACE. In case of serial port communication, the monitor services are directly available from the host.

Daughter-boards containing up to 16 Texas TMS320C40 floating-point processors can be installed in CNN-ACE, each of them with "identity" and dedicated program and data memory of unique address space. After power-up test, programs and data are downloaded from the host through the transparent (SCSI or serial) communication channel. To initialise and implement the RESET-RUN-STOP-CONTINUE commands and the corresponding RESET-RUN-STOP-WAIT states, first a standard initialiser program written in TMS320C4x Assembly Language is downloaded, run and goes into WAIT state. Application programs written in C or Assembly Language can be downloaded in this state and started with indirect branch from the START_ADDRESS loaded. A library of interprocessor communication and DMA coprocessor routines are available for application program development.

The CNN Simulator comes together with a library of on-board CNN routines organised under a switch controlled from the host. They are written in C and for the sake of speed the most time-intensive ones are manually optimised and checked for pipeline conflicts in assembly level. The library items can be grouped into i) template routines to calculate the CNN differential equation, ii) interprocessor communication for exchanging image segment frames, iii) DRAM-SRAM-caching DMA routines, iv) (floating) number format conversions, v) local control and status report routines. Among the template routines, in addition to the generic ones, several optimised routines are available for small, most frequently used template sizes.

4.1. Application example

For illustrating the application of ACE, Fig. 4 shows the spatio-temporal evolution of the scroll wave in a 40*40*3 CNN modelling a 2D array of Chua's circuits by three layers of nonlinear CNN cells [9]. In Fig. 5, a snapshot of the twisted scroll wave in an inhomogeneous 3D array of Chua's circuits - also based on [9] - consisting of 40*40*(40*3)=192000 cells, is shown. In both cases the 32 bit floating point accuracy was crucial in achieving correct results. The computation time of the nonlinear three-layer templates in each 'C40 processor was 7.7µsec/cell/iteration.

5. Conclusion

To enhance the speed, accuracy and capacity of CNN calculations and simulation a floating point multiprocessor accelarator engine CNN-ACE has been built. In maximum configuration it can accomodate 16 TMS320C40 DSPs and 512 Mbyte RAM assembled on 8 daughter-boards of a SCSI accessed mother-board. In addition to its large virtual processor space the most critical parameter of the CNN-ACE is the computational speed which, for 3*3 linear templates, is ~2.87 μsec/cell/iteration/'C40processor.

Although the CNN-ACE architecture was designed and programmed to be a CNN emulator engine, other types of algorithms, general purpose calculations are supported on the system, as well.

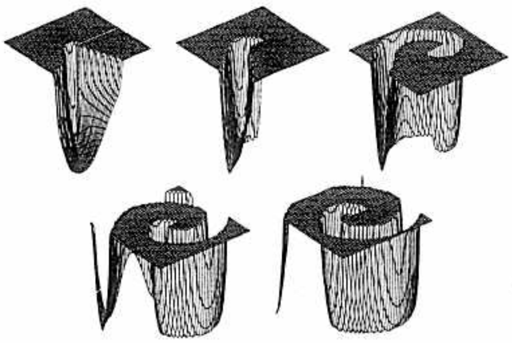


Figure 4: Scroll wave in the 2D Chua's circuit array after 11, 17, 27, 35 and 39τ.

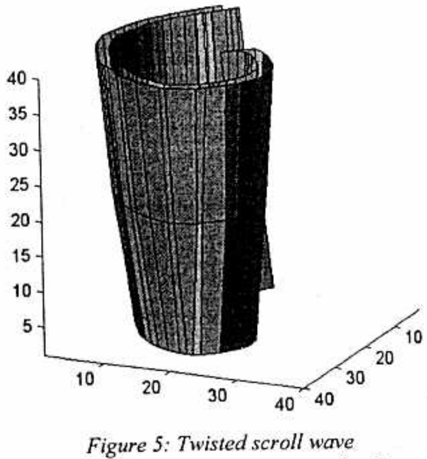


Figure 5: Twisted scroll wave in the 3D Chua's circuit array after 37τ

6. Acknowledgement

This work has also been supported by the research grants No.7308 and T012860 of the National Research Fund of Hungary (OTKA)

7. References

[1] L.O.Chua and L.Yang: "Cellular neural networks: Theory" and "Cellular neural networks: Applications", *IEEE Trans. on Circuits and Systems*, Vol.35, pp. 1257-1290, 1988.

[2] T.Roska and L.O.Chua: "The CNN Universal Machine: an analogic array computer", *IEEE Transactions on Circuits and Systems-II*, Vol.40, pp. 163-173, March 1993.

[3] T.Roska, G.Bártfai, P.Szolgay, T.Szirányi, A.Radványi, T.Kozek, Zs. Ugray and A Zarándy: "A digital multiprocessor Hardware Accelerator Board for Cellular Neural Networks: CNN-HAC", *Int. J. of Circuit Theory and Applications*, Vol.20, No.5, pp. 589-599, 1992.

[4] T.Roska, L. Kék (editors): "Analogic CNN program library, Version 6.2", *DNS-7-1995*. MTA-SzTAKI Budapest, 1995.

[5] *CNN Workstation Toolkit User's Manual*, MTA-SzTAKI, Budapest, 1995.

[6] *TEXAS Instruments TMS320C4X User's Guide*, 1992.

[7] P.Szolgay, G.Vörös:"Transient response computation of a mechanical vibrating system using Cellular Neural Networks", *Proc.of CNNA'94*, pp.321-326, Rome, 1994.

[8] T.Roska, P.Szolgay, A Zarándy, P.L.Venetianer, A.Radványi and T.Szirányi:"On a CNN chip-prototyping system" Sec.2.4, *Proc.of CNNA'94*, pp.378-379, Rome, 1994.

[9] L. Pivka, "Autowaves and Spatio-Temporal Chaos in CNNs-Part I&II: A Tutorial", *IEEE Trans. on Circuits and Systems*, Vol .42(I), pp. 638-649 & 650-664, October 1995